基于 OpenMP 的高效多子阵合成孔径声纳距离多普勒成像算法 *

钟何平，黄　攀，唐劲松

(海军工程大学 海军水声技术研究所, 武汉 430033)

摘　要：为充分利用多核 CPU 计算资源解决多子阵合成孔径声纳成像效率低的问题，提出了一种共享内存环境下的距离多普勒成像算法并行解决方案。在分析多子阵合成孔径声纳距离多普勒成像算法并行性的基础上，对算法中预处理、距离向脉冲压缩、固定相位补偿、距离徙动校正和方位向脉冲压缩进行了 OpenMP 并行化设计，充分利用多核 CPU 计算资源实现了大数据量合成孔径声纳图像快速重构。对实测数据的成像实验结果表明，并行成像算法加速比高达 19.86，满足实时合成孔径声纳系统成像需求。

关键词：合成孔径声纳；距离多普勒成像算法；并行计算；共享内存；OpenMP

中图分类号：TN958　　doi: 10.3969/j.issn.1001-3695.2017.10.1015

Efficient range doppler imaging algorithm for multi-array synthetic aperture sonar based on OpenMP

Zhong Heping, Huang Pan, Tang Jinsong

(*Naval Institute of Underwater Acoustic Technology*, *Naval University of Engineering*, *Wuhan* 430033, *China*)

Abstract: An efficient range Doppler imaging algorithm for multi-array synthetic aperture sonar (SAS) in shared memory environment based on OpenMP is proposed, which solves the imaging problem of low efficiency by multi-core CPU resources. Considered the parallel characteristic of range Doppler algorithm for multi-array SAS, The signal processing process have been parallelized such as preprocessing, range compression, phase compensation, range cell migration compensation and azimuth compression based on OpenMP, which greatly enhances the reconstruction process of SAS image by multi-core CPU resources. The imaging test performed on real data shows that the speedup of the parallel imaging algorithm is up to 19.86 times, which meets the demand for real-time processing of SAS imaging.

Key Words: synthetic aperture sonar; range Doppler imaging algorithm; parallel computing; shared memory; OpenMP

0　引言

多子阵合成孔径声纳(synthetic aperture sonar，SAS)是一种高分辨率成像声纳[1~3]，其显著特点是方位向分辨率与距离无关，并且高于常规侧扫声纳 1~2 个数量级，在水下小目标探测应用方面具有重要用途[4~6]。多子阵 SAS 得以应用的一个重要前提就是实现合成孔径声纳图像的快速重建[7,8]，而距离多普勒成像算法(range Doppler algorithm，RDA)是一种经典的合成孔径成像算法，兼具模块化处理能力和一维操作简便性，至今仍被广泛应用于多种合成孔径成像场景[9]。随着合成孔径声纳系统成像分辨率的不断提高和测绘带宽度的不断加大，用于成像的原始数据量不断加大，严重影响了合成孔径声纳系统的成像效率。SAS 实时成像在硬件上可采用专用信号处理机[10]，但存在研发周期长、价格高和不易拓展等缺点。为了提升计算性能，集群系统[8,11]也用在了 SAS 成像，获得了实时处理结果，但采用集群方式计算存在计算设备体积大、功耗大，不能满足水声设备的小体积、低功耗，运输方便的要求。

近年来，随着多核技术不断成熟，共享内存架构[12]已成为各类型计算节点的标准配置，并且共享内存并行程序拓展性好，其性能与单核计算速度和核数成正比，可在不同平台上可直接运行。本文在此基础上提出了一种共享内存环境下的多子阵 SAS 距离多普勒快速成像算法，有效提升了合成孔径声纳成像效率，满足实时合成孔径声纳成像需求。

1　共享内存并行

1.1　OpenMP 并行模型

由于多核技术的发展，采用多核 CPU 进行并行计算来提高计算效率已被广泛采用。多核 CPU 的多个线程在物理上是并行的，其编程属于共享存储的编程模型。OpenMP 是共享存储器模型的并行编程标准，它是为共享存储环境编写并行程序设计

基金项目：国家自然科学基金资助项目（61671461，41304015）；中国博士后科学基金资助项目（2015M582813）

作者简介：钟何平（1983-），男，湖北钟祥人，讲师，博士，主要研究方向为干涉信号处理、并行计算（zheping525@sohu.com）；黄攀（1986-），男，博士研究生，主要研究方向为干涉合成孔径声纳图像处理；唐劲松（1964-），男，研究员，博士，主要研究方向为合成孔径声纳、水声通信.

的一个应用程序接口，目前支持 Fortran 和 C/C++，其标准中包括一套编译指导语句及一个支持函数库。OpenMP 在并行执行程序时，采用的是“Fork/Join”方式，其并行执行过程如图 1 所示。程序开始时只有一个主线程，程序中的串行区域都由主线程执行，并行部分通过派生其他线程来共同完成。从图 1 中可以看出，OpenMP 程序并行结束之前是不能执行串行部分的，这就是标准的 Fork/Join 并行模式。

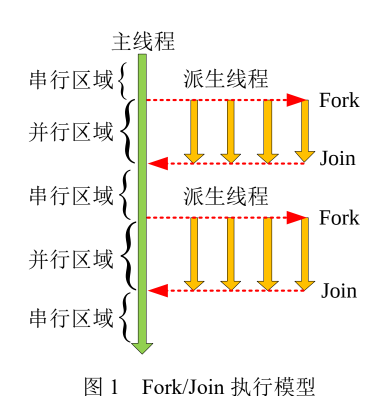


图 1　Fork/Join 执行模型

1.2　并行计算效率

加速比是衡量并行算法性能的一个重要指标，它定量描述了一个程序实现并行化后，由于执行时间的减少而获得的性能。在处理器资源独享的前提下，假设某个应用程序的执行时间为 T_s，而该程序并行化后，在 P 个处理器上并行执行所需的时间为 T_p，则该并行程序在该计算机上的加速比 S_p 定义为：$S_p = T_S / T_p$。理论上，S_p 的值越大，并行系统获得的效率越高。加速比通常小于 CPU 的总核数。因此，加速比一般要求向 CPU 核数靠近。

并行算法的效率高并不等于处理器的利用率很高，为此引入算法效率的概念。效率 E_p 度量了每个处理器有效使用部分占总的执行时间比，通常定义位：$E_p = S_p / P$。

并行算法的效率理想值为 1，但一般情况下都小于 1，主要是因为程序的并行化程度不高，而且存在通信以及同步等时间开销所致。

1.3　测试平台

为测试并行成像算法性能，本文选用了一台笔记本和一个刀片计算节点，具体配置如下：

笔记本：处理器 Intel(R) Core(TM) i5-4200H CPU@2.80 G B 4 核，显卡 NVIDIA GeForce GTX 950 MB，内存 4 GB，操作系统 Windows 7 旗舰版 64 位，软件环境开发环境：Visual Studio 2008。

刀片计算节点：处理器 Intel(R) Xeon(R) CPU E5-2690 v2@3.0 GB 2 处理器 40 核，内存 32 GB，操作系统 Windows Server 2008 64 位，软件环境开发环境：Visual Studio 2008。

2　算法描述

2.1　多子阵 SAS 距离多普勒成像算法

距离多普勒成像算法利用距离相同而方位不同的散射点在距离多普勒域具有相同的能量轨迹，通过插值解二维耦合，将成像的二维处理转换成两个一维处理级联的形式，实现距离向处理与方位向处理的分离。多子阵 SAS 常用成像方法就是将多子阵回波信号转换为单子回波信号形式，然后调用单阵距离多普勒成像算法实现图像重构。图 2 给出了多子阵 SAS 距离多普勒成像算法的基本流程。图中 FFT 和 IFFT 分别表示傅里叶变换和傅里叶逆变换。

从信号处理流程上看，多子阵 SAS 距离多普勒成像主要包括预处理、距离向脉冲压缩、固定相位补偿和方位向脉冲压缩四个步骤。在进行方位向脉冲压缩，回波信号变换到距离多普勒域，距离向和方位向信号不能解耦和时，需要利用耗时的插值操作来完成距离向和方位向解耦。

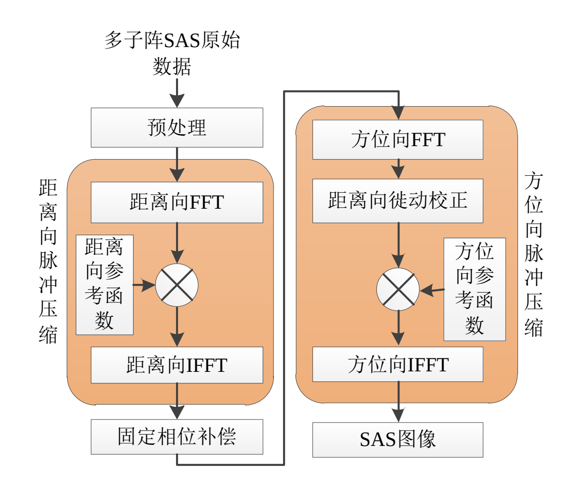


图 2　距离多普勒算法流程

2.2　算法并行性能分析

2.2.1 预处理

SAS 原始数据是按照采集脉冲顺序依次进行存储，脉冲内部按照通道编号顺序进行存储，对于一幅 SAS 原始回波数据，其距离向点数为 NR，方位向点数为 $NAO = NPluse \cdot NC$，其中 $NPluse$ 和 NC 分别表示脉冲个数和接收阵个数。预处理主要是对采集的原始数据进行数据类型转换和进行有效子阵数据截取，其中数据类型转换是将采样点由 short 类型转换为 float 类型，其并行度为 $NAO \bullet NR$。有效子阵截取是根据脉冲重复间隔 PRI、载体速度 V 和子阵长度 D 之间的关系 $PRI \bullet V = D \bullet M / 2$ 来确定用于成像的有效子阵个数 M，然后对原始数据每个脉冲截取前 M 个子阵数据组成新的数据块，截取后的数据块大小为 $NA \bullet NR$，其中 $NA = NPluse \cdot M$。由于截取过程是以单个脉冲数据为对象进行的数据搬移操作，所以并行度为 $NPluse$。

2.2.2 距离向脉冲压缩

预处理后的数据进行距离向脉冲压缩时，主要包含距离向 FFT、与距离向参考函数相乘和距离向 IFFT 三个处理步骤。由于预处理后的数据是按照脉冲顺序进行存储的，在进行距离向 FFT 变换时，不同方位向上距离线数据之间是相互独立的（即图像中两行数据是相互独立的），可同时进行 FFT 变换，其并

行度为 NA。距离向参考函数为

$$H(f_r)=\exp\left(j\pi f_r^2/\mu\right) \tag{1}$$

其中：f_r 表示距离向基带频率；μ 为信号调频斜率，它与每一距离线上数据相乘时是相互独立的，而不同距离线上数据相乘也是相互独立的，因此该处理步骤并行度为 $NA \bullet NR$。与 FFT 变换相似，距离向 IFFT 变换的并行度也为 NA。

2.2.3 固定相位补偿

固定相位补偿是将距离向脉冲压缩后的多子阵回波变为单子阵回波形式，以便利用单阵 SAS 距离多普勒成像算法进行处理。固定相位补偿精度直接关系到最终成像结果的分辨率，不同学者针对具体成像条件下的相位补偿方法进行了深入研究[13]。引入停走停近似修正后的偏置相位中心近似，且停走停的修正只考虑距离空变情形下的固定相位补偿公式为

$$H(\Delta h_i,r)=\exp\left\{j\frac{2\pi f_0}{rc}\left(v\frac{r}{c}+\frac{\Delta h_i}{2}\right)^2\right\} \tag{2}$$

其中：f_0 为中心频率；Δh_i 为第 i 个接收子阵与发射阵方位向等效相位中心间隔；c 为声速；v 为载体速度；r 为斜距。从相位补偿因子可看出其为 Δh_i 和 r 两个变量的函数，每一点的补偿过程是相互独立的，因此该处理步骤并行度为 $NA \bullet NR$。

2.2.4 方位向脉冲压缩

方位向脉冲压缩主要包括方位向 FFT、距离向徙动校正、与方位向参考函数相乘和方位向 IFFT。由于声纳数据是按照距离向排列的，在进行方位向处理前需要进行矩阵转置操作，方位向处理完成后再进行一次矩阵转置操作变回为先前排列方式。在进行方位向 FFT 变换时，不同距离向上方位线数据之间是相互独立的，可同时进行 FFT 变换，其并行度为 NR。

距离向徙动校正是在距离多普勒域采用插值实现最近距离相同而方位不同点目标能量曲线的统一校正，其效率与插值核有关，通常 8 点插值核已经能够满足要求。原始数据变换至距离多普勒域后，需插值校正的时间偏移量是一个与 f_a 和 r 相关的二维空变函数：

$$\Delta\tau(f_a,r)=\frac{2r}{c}\left(1-\frac{1}{\sqrt{1-c^2 f_a^2/\left(4v^2 f_0^2\right)}}\right) \tag{3}$$

其中：c 为声速；v 为载体速度；r 为斜距；f_0 为中心频率；f_a 为方位向频率。距离多普勒域中每一点的校正过程是相互独立的，因此该步骤的并行度为 $NA \bullet NR$。

在距离多普勒域中，方位向参考函数为

$$H(f_a,r)=\exp\left(j\frac{2\pi r}{vc}\sqrt{4v^2 f_0^2-c^2 f_a^2}\right) \tag{4}$$

可见方位向参考函数也为一个与 f_a 和 r 相关的二维空变函数，它与距离多普勒域中每一点数据相乘是相互独立的，因此该处理步骤并行度为 $NA \bullet NR$。

方位向脉冲压缩的最后一步为 IFFT 变换。与 FFT 变换类似，方位向 IFFT 变换的并行度也为 NR。成像算法各处理步骤的并行度如表 1 所示。

表 1　成像算法各处理步骤并行度

处理步骤名称	子步骤名称	并行处理粒度	并行度
预处理	数据类型转换	数据点	$NA \bullet NR$
	有效子阵截取	脉冲块	$NPluse$
距离向脉冲压缩	距离向 FFT 和 IFFT	距离线数据	NA
	距离向参考函数相乘	数据点	$NA \bullet NR$
固定相位补偿	固定相位补偿	数据点	$NA \bullet NR$
方位向脉冲压缩	方位向 FFT 和 IFFT	方位线数据	NR
	距离向徙动校正	数据点	$NA \bullet NR$
	方位向参考函数相乘	数据点	$NA \bullet NR$

2.3　共享内存环境下并行算法设计

2.3.1 预处理

预处理主要包括数据类型转换和有效子阵截取。为避免地址计算，前者采用一重 for 循环实现，后者采用两重 for 循环实现，其中后者外层 for 针对脉冲数进行循环，两者都通过 parallel for 指令进行并行化。有效子阵截取步骤针对外层脉冲数循环进行并行，伪代码如下所示。

```
Preprocessing(A,B, NumOfPluse, Nc)
A:complex<short>类型原始数据
B:complex<float>类型数据
NumOfPluse:脉冲个数
Nc:有效子阵个数
parallel for i=1 to length[A]
B[i] ← A[i] / 32768.0
parallel for i=1 to NumOfPluse
for j=1 to Nc
提取B中第i个脉冲第j个子阵数据构成新数据块，舍去多余子阵数据。
```

2.3.2 距离向脉冲压缩

在进行距离压缩前，首先需要计算出距离向参考函数。为能够与 FFT 变换结果直接相乘，在计算参考函数时，f_r 的取值范围设定为 $\left[0,rF_s\right]$ 来避免频谱搬移带来的额外运算量，其中 rF_s 表示距离向采样频率。距离向脉冲压缩通过 parallel for 指令对外层行循环进行并行化，每一行数据依次进行 FFT 变换、与距离向参考函数点乘和 IFFT 变换，并行完成距离向脉冲压缩。

2.3.3 固定相位补偿

根据固定相位补偿公式可知，不同脉冲数据的补偿因子相同，并且补偿过程也相同。为避免相位补偿因子重复计算，本文先开辟单个脉冲数据大小的临时空间存储相位补偿因子，然后计算出补偿因子数组，最后与声纳数据相乘完成多子阵回波向单阵回波的转换。初始化相位补偿因子过程中，将距离向采样点数循环放在外层，子阵个数循环放在内层。在与声纳数据

对应相乘过程中，将脉冲数循环放在外层，通过 parallel for 指令实现并行固定相位补偿。

2.3.4 方位向脉冲压缩

由于固定相位补偿后数据是按照先距离向后方位向排列，进行方位向 FFT 变换前，首先需要进行数据重组。考虑到不同距离向数据 FFT 变换满足独立性，为同时进行 FFT 变换，本文在每个线程内部开辟长度为 *NA* 的临时数据空间，将原始内存离散的方位向数据搬移到内存连续的临时数据空间，完成 FFT 变换后再将数据搬移至原来位置，同时删除临时内存空间。

接下来是进行距离徙动校正。由于不同行（这里表示方位向频率）的校正过程是独立的，本文对外层行循环直接采用 parallel for 指令进行并行。因插值过程中要使用固定偏移量位置相邻数据，线程内部访问数组时要进行严格的下标有效性判断。

最后一步是与方位向参考函数相乘并进行 IFFT 变换，这一步不同距离向的处理过程是相互独立的，但是不同距离向的方位向参考函数不同。为使不同距离向数据同时与方位向参考函数相乘并进行 IFFT 变换，在每个线程内部开辟长度为 *NA* 的临时数据存储空间，将原始内存离散的方位向数据搬移到内存连续的临时空间，然后计算方位向参考函数并与对应方位向数据相乘，最后进行 IFFT 变换后再将数据搬移至原来位置，完成方位向脉冲压缩，实现最终 SAS 图像重构。

3 实验结果

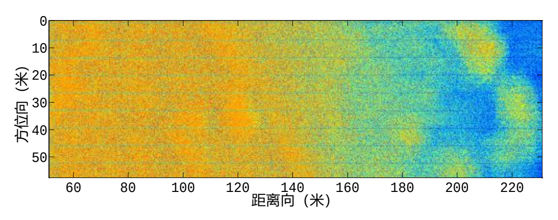
为验证共享内存环境下的多子阵 SAS 距离多普勒成像算法效率，本文在 1.3 节所示两种并行计算平台上测试了 CPU 串行计算方法和基于 OpenMP 共享内存并行计算方法的成像效率。不同平台上测试时，使用同一程序，程序启动时，首先自动获取当前计算平台上核数，并将其设置为计算核数，然后再进行成像处理。成像过程中，距离徙动校正采用 8 点的 sinc 插值方法，由于预处理所占时间少，在效率分析时忽略不计。

实验所用数据是 2010 年 7 月在某内陆湖进行干涉合成孔径声纳海试样机实验中获取的，具体实验参数如表 2 所示。实验所用数据块原始回波信号幅度如图 3(a)所示。该数据块包含 4 个合成孔径长度共计 72 个脉冲数据，距离向点数为 9 600，方位向点数为 3 456，获取时间为 23.04 s。由于相邻数据块之间存在 1 个合成孔径长度的数据重叠，所以用于成像的时间上限仅为 17.28 s。信号处理过程中，有效子阵数据个数为 40，成像结果数据方位向点数变为 2 880。本文首先在两种并行计算环境下进行了串行和并行成像实验，图 3(b)和 (c)分别是成像程序串行运行和并行运行所得的成像结果。从成像效果图来看，两者相同，验证了并行成像程序功能的正确性。

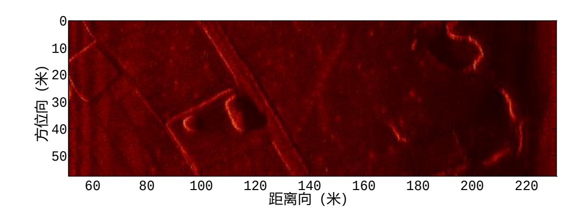
两种并行计算平台下，串行与并行 SAS 距离多普勒成像算法效率比较结果如表 3 所示。在计算平台 A 上，串行算法成像时间为 16 731 ms，并行算法成像时间为 6 534 ms，加速比为 2.56，计算效率为 0.64。成像算法中距离向脉冲压缩、固定相位补偿、方位向 FFT、距离徙动校正和方位向 IFFT 各处理步骤加速比分别为 2.39、2.40、2.57、2.53、2.76 和 2.56。在计算平台 B 上，串行算法成像时间为 19 244 ms，并行算法成像时间仅为 969 ms，加速比高达 19.86，计算效率为 0.50。成像算法中距离向脉冲压缩、固定相位补偿、方位向 FFT、距离徙动校正和方位向 IFFT 各处理步骤加速比分别为 13.55、9.41、18.01、23.47 和 27.83。从两组实验对比结果可看出，计算平台 B 上加速比高，主要是通过增加计算核数获得的，但与计算平台 A 相比，并行效率较低，与原始数据有效部分获取时间 17.28 s 相比，满足实时合成孔径声纳系统成像需要。

表 2 实验系统参数

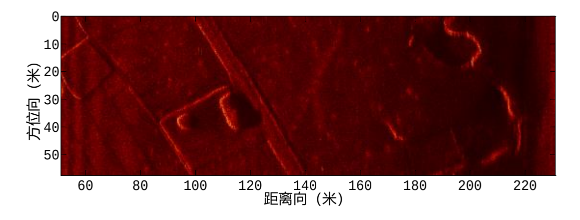
参数	数值	参数	数值
载频(kHz)	150	脉冲重复时间(ms)	320
带宽(kHz)	20	子阵长度(m)	0.08
脉宽(ms)	20	子阵个数	48
声速(m/s)	1448	载体速度(m/s)	2.5



(a)原始回波信号



(b) 串行成像结果



(c) 并行成像结果

图 3 实测数据成像实验

4 结束语

本文在分析多子阵 SAS 距离多普勒成像算法并行性特点的基础上，利用 OpenMP 并行工具在共享内存环境下提出了一种快速合成孔径声纳距离多普勒成像算法。针对多子阵 SAS 距离多普勒成像算法中的预处理、距离向脉冲压缩、固定相位补偿和方位向脉冲压缩关键处理步骤完成了共享内存环境下的并行化设计，并在不同并行计算平台上采用实测数据进行了成像性能和效率对比测试。实验结果表明，所提并行成像算法加速

比随着计算平台核数增加而变大，当计算核数较多时，所提并行成像算法满足实时成像需求。

表 3　不同计算方法效率比较/ms

计算平台	成像方法	距离向脉冲压缩	固定相位补偿	方位向 FFT	距离徙动校正	方位向脉冲压缩	总时间	效率
A	串行	3360	231	2840	6216	4084	16731	0.64
	并行	1405	96	1105	2455	1482	6543	
	加速比	2.39	2.40	2.57	2.53	2.76	2.56	
B	串行	3564	480	3043	7370	4787	19244	0.50
	并行	263	51	169	314	172	969	
	加速比	13.55	9.41	18.01	23.47	27.83	19.86	

注:计算平台 A 表示笔记本,计算平台 B 表示刀片计算节点。

参考文献：

[1] Sæbø T O, Synnes S A V, Hansen R E. Wideband interferometry in synthetic aperture sonar [J]. IEEE Trans on Geoscience and Remote Sensing, 2013, 51 (8): 4450-4459.

[2] Hayes M P, Gough P T. Synthetic aperture sonar: a review of current status [J]. IEEE Journal of Oceanic Engineering, 2009, 34 (3): 207-224.

[3] Zhang X B, Tang J S, Zhong H P. Multireceiver correction for the chirp scaling algorithm in synthetic aperture sonar [J]. IEEE Journal of Oceanic Engineering, 2014, 39 (3): 472-481.

[4] Tian Z, Tang J S, Zhong H P, et al. Extended range Doppler algorithm for multiple-receiver synthetic aperture sonar based on exact analytical two-dimensional spectrum [J]. IEEE Journal of Oceanic Engineering, 2016, 41 (1): 164-174.

[5] Piper J E, Commander K W, Thorsos E I, et al. Detection of buried targets using a synthetic aperture sonar [J]. IEEE Journal of oceanic engineering, 2002, 27 (3): 495-504.

[6] Hansen R E, Sæbø T O, Callow H J, et al. Interferometric synthetic aperture sonar in pipeline inspection [C]// Proc of OCEANS 2010 IEEE-Sydney. [S. L.] : IEEE Press, 2010: 1-10.

[7] 杨敏，宋士林，徐栋，等. 合成孔径声纳技术以及在海底探测中的应用研究 [J]. 海洋技术学报, 2016, 35 (2): 51-55.

[8] Riyait V S, Lawlor M A, Adams A E, et al. Real-time synthetic aperture sonar imaging using a parallel architecture [J]. IEEE Trans on Image Processing, 1995, 4 (7): 1010-1019.

[9] 江泽林，刘维，李保利，等. 基于集群的高频合成孔径声呐并行处理方法 [J]. 应用声学, 2011, 30 (3): 167-176.

[10] Tian Z, Tang J S, Zhong H P, et al. Extended range Doppler algorithm for multiple-receiver synthetic aperture sonar based on exact analytical two-dimensional spectrum [J]. IEEE Journal of Oceanic Engineering, 2016, 41 (1): 164-174.

[11] 陈东升，刘维，刘纪元，等. 基于 PowerPC 的合成孔径声纳实时信号处理系统 [J]. 微计算机应用, 2008, 29 (7): 6-10.

[12] Li B, Liu W, Liu J, et al. Real-time implementation of synthetic aperture sonar imaging on high performance clusters [C]// Proc of the 11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel//Distributed Computing. 2010: 89-92.

[13] Dagum L, Menon R. OpenMP: An industry standard API for shared-memory programming [J]. IEEE Computational Science and Engineering, 1998, 5 (1): 46-55.

[14] 杨海亮，张森，唐劲松，等. 一种精确的多接收阵合成孔径声纳距离多普勒成像算法 [J]. 数据采集与处理, 2010, 25 (3): 313-317.